

**NPS ARCHIVE**  
**1968**  
**GREGOR, R.**

OPTIMUM SHIP ROUTING BY THE  
METHOD OF STEEPEST ASCENT

by

Richard Allen Gregor

Lt. Col.  
NAVAL POSTGRADUATE SCHOOL  
MONTEREY, CALIF. 93940

# UNITED STATES NAVAL POSTGRADUATE SCHOOL



## THESIS

OPTIMUM SHIP ROUTING BY THE  
METHOD OF STEEPEST ASCENT

by

Richard Allen Gregor

September 1968

This document is the property of the U.S. Navy and is loaned to your organization. It and its contents are not to be distributed outside your organization without the approval of the U.S. Naval Postgraduate School.

[Add to Shopping Cart](#)[Save for Later](#)

## Technical Reports Collection

Citation Format: Full Citation (1F)

**Accession Number :**

AD0842884

**Citation Status:**

Active

**Citation Classification:**

Unclassified

**Field(s) & Group(s):**

120500 - COMPUTER PROGRAMMING AND SOFTWARE

**Corporate Author:**

NAVAL POSTGRADUATE SCHOOL MONTEREY CA

**Unclassified Title:**

Optimum Ship Routing by the Method of Steepest Ascent.

**Title Classification:**

Unclassified

**Descriptive Note:**

Master's thesis,

**Personal Author(s):**

Gregor, Richard Allen

**Report Date:**

01 Sep 1968

**Media Count:**

LIBRARY  
NAVAL POSTGRADUATE SCHOOL  
MONTEREY, CALIF. 93940

OPTIMUM SHIP ROUTING BY THE  
METHOD OF STEEPEST ASCENT

by

Richard Allen Gregor  
Lieutenant, United States Navy  
B.S., Naval Academy, 1961

NO FORN

Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN MATHEMATICS

FROM THE

NAVAL POSTGRADUATE SCHOOL  
September 1968



[Add to Shopping Cart](#)[Save for Later](#)

## Technical Reports Collection

Citation Format: Full Citation (1F)

**Accession Number :**

AD0842884

**Citation Status:**

Active

**Citation Classification:**

Unclassified

**Field(s) & Group(s):**

120500 - COMPUTER PROGRAMMING AND SOFTWARE

**Corporate Author:**

NAVAL POSTGRADUATE SCHOOL MONTEREY CA

**Unclassified Title:**

Optimum Ship Routing by the Method of Steepest Ascent.

**Title Classification:**

Unclassified

**Descriptive Note:**

Master's thesis,

**Personal Author(s):**

Gregor, Richard Allen

**Report Date:**

01 Sep 1968

**Media Count:**

53 Page(s)

**Cost:**

\$7.00

**Report Classification:**

Unclassified

**Descriptors:**

(\*MARINE TRANSPORTATION, , SCHEDULING), CARGO SHIPS, WATER TRAFFIC, NORTHERN HEMISPHERE, PACIFIC OCEAN, OCEAN WAVES, GEODESICS, COMPUTER PROGRAMS, NONLINEAR DIFFERENTIAL EQUATIONS, CALCULUS OF VARIATIONS, NUMERICAL ANALYSIS, OPTIMIZATION, COSTS, THESES

**Identifiers:**

OPTIMAL ROUTING THEORY.

**Abstract:**

With the advent of the high speed digital computer, many problems heretofore considered unsolvable for all practical purposes are now well within the reach of the applied mathematician. One such problem is the routing of a ship through a time dependent ocean wave field, from one point on the earth's surface to another, so as to minimize a cost function of the form  $g(x,y,t,u)$ . This paper considers a numerical solution to the above problem. The technique to be employed is known as the method of steepest ascent and is attributed to Arthur E. Bryson and Walter F. Denham. Although the computer program as given in the Appendix is written specifically for a VC2AP3 class vessel operating in a described area of the North Pacific Ocean, it can be readily modified to accommodate any type vessel operating in the Northern Hemisphere. (Author)

**Abstract Classification:**

Unclassified

**Distribution Limitation(s):**

01 - APPROVED FOR PUBLIC RELEASE

**Source Code:**

251450

**Document Location:**

DTIC

**Change Authority:**

ST-A USNPS LTR, 23 SEP 71

---

NPS Archive  
1968  
Gregor, R

~~XX  
5/8~~  
ABSTRACT

With the advent of the high speed digital computer, many problems heretofore considered unsolvable for all practical purposes are now well within the reach of the applied mathematician. One such problem is the routing of a ship through a time dependent ocean wave field, from one point on the earth's surface to another, so as to minimize a cost function of the form  $g(x,y,t,u)$ .

This paper considers a numerical solution to the above problem. The technique to be employed is known as the method of steepest ascent and is attributed to Arthur E. Bryson and Walter F. Denham [1]. Although the computer program as given in the Appendix is written specifically for a VC2AP3 class vessel operating in a described area of the North Pacific Ocean, it can be readily modified to accommodate any type vessel operating in the Northern Hemisphere.



## TABLE OF CONTENTS

SECTION	PAGE
I Statement of the Problem	5
II Abstract of the Theory	9
III Development of the Theory	11
IV The Numerical Solution	19
V Conclusions	23
BIBLIOGRAPHY	25
APPENDIX	26

#### ACKNOWLEDGMENTS

I am indebted to Professor F. D. Faulkner, who stimulated interest in the problem investigated, to my thesis advisor Professor W. E. Bleick for his guidance, and to the Fleet Numerical Weather Central for the use of the Control Data Corporation 6500 Digital Computer.

## I. Statement of the Problem

The basic problem to be developed and solved in this paper is the following: A VC2AP3 vessel, located at latitude  $40^{\circ}\text{N}$  longitude  $154^{\circ}\text{E}$ , is to transit a time dependent ocean wave height and wave direction field to latitude  $38^{\circ}\text{N}$  longitude  $123^{\circ}\text{W}$  so as to minimize a cost function of the form  $g(x,y,u,t)$ .

A south-polar stereographic projection of the Northern Hemisphere upon a plane passing through the circle of  $60^{\circ}\text{N}$  latitude is used to establish a rectangular coordinate system, with the OX and OY axes parallel to the projections of the meridians of  $10^{\circ}\text{E}$  and  $100^{\circ}\text{E}$  longitude, respectively. Then a  $63 \times 63$  grid is constructed along the OX and OY axes, with  $x = y = 31$  defining the projection of the North Pole. At each point of this grid, actual wave heights and wave directions taken from the files of the U.S. Navy Fleet Numerical Weather Central, Monterey, California, are recorded. The map scale factor MSF, defined as the ratio of a differential distance in the OXY plane to the corresponding differential distance on the earth's surface, is

$$\text{MSF} = [973.75 + (x-31)^2 + (y-31)^2] / 1043.6.$$

Germane to the solution of the above problem is the elliptical polar velocity diagram of Figure 1, giving the ship's speed  $v$  as a function of the angle  $\theta$  between the ship's heading and the wave direction. R. W. James [6] gives empirical curves for the speed of a VC2AP3 class vessel in head waves  $v_h$ , beam waves  $v_b$ , and following waves  $v_f$  as a

function of wave height. Bleick [2] showed through a least squares analysis that these three speeds can be represented closely by the 4-parameter formula

$$v = C_4 (H + C_1)^{C_2} \exp(-C_3 H) ,$$

where  $H$  is the wave height.

Thus, the projected speed of the ship at any point on the grid is

$$V(x, y, \theta, t) = v(H, \theta) \text{MSF} ,$$

where  $v(H, \theta)$  is the ship's speed of the polar velocity diagram of Figure 1, and  $H(x, y, t)$  is the interpolated value of the Fleet Numerical Weather Central wave height data. If we define  $K(x, y, t)$  to be the stereographic grid wave direction measured counterclockwise from the OX axis, and  $u(t)$  to be the ship's heading as measured counterclockwise from the OX axis, then

$$[v \sin(u-K)/b]^2 + \{[c+v \cos(u-K)]/a\}^2 = 1 \quad \text{or}$$

$$v(H, \theta) = \frac{b(a^2 - c^2)}{a} / \left[ \frac{bc}{a} \cos \theta + \sqrt{b^2 \cos^2 \theta + (a^2 - c^2) \sin^2 \theta} \right]$$

where  $\theta = u - K$ ,  $a$  and  $b$  are the semi-principal axes of Figure 1, and  $c$  is the distance to the eccentric pole  $o$ . We note at this point that an elliptical polar velocity diagram, such as Figure 1, must be specified for each wave height, and that  $a$ ,  $b$ ,  $c$  are functions of  $H(x, y, t)$ . Thus, we can write the equations of motion of the ship's projection in the OXY plane as:

$$\dot{x}(t) = V \cos u(t)$$

$$\dot{y}(t) = V \sin u(t)$$

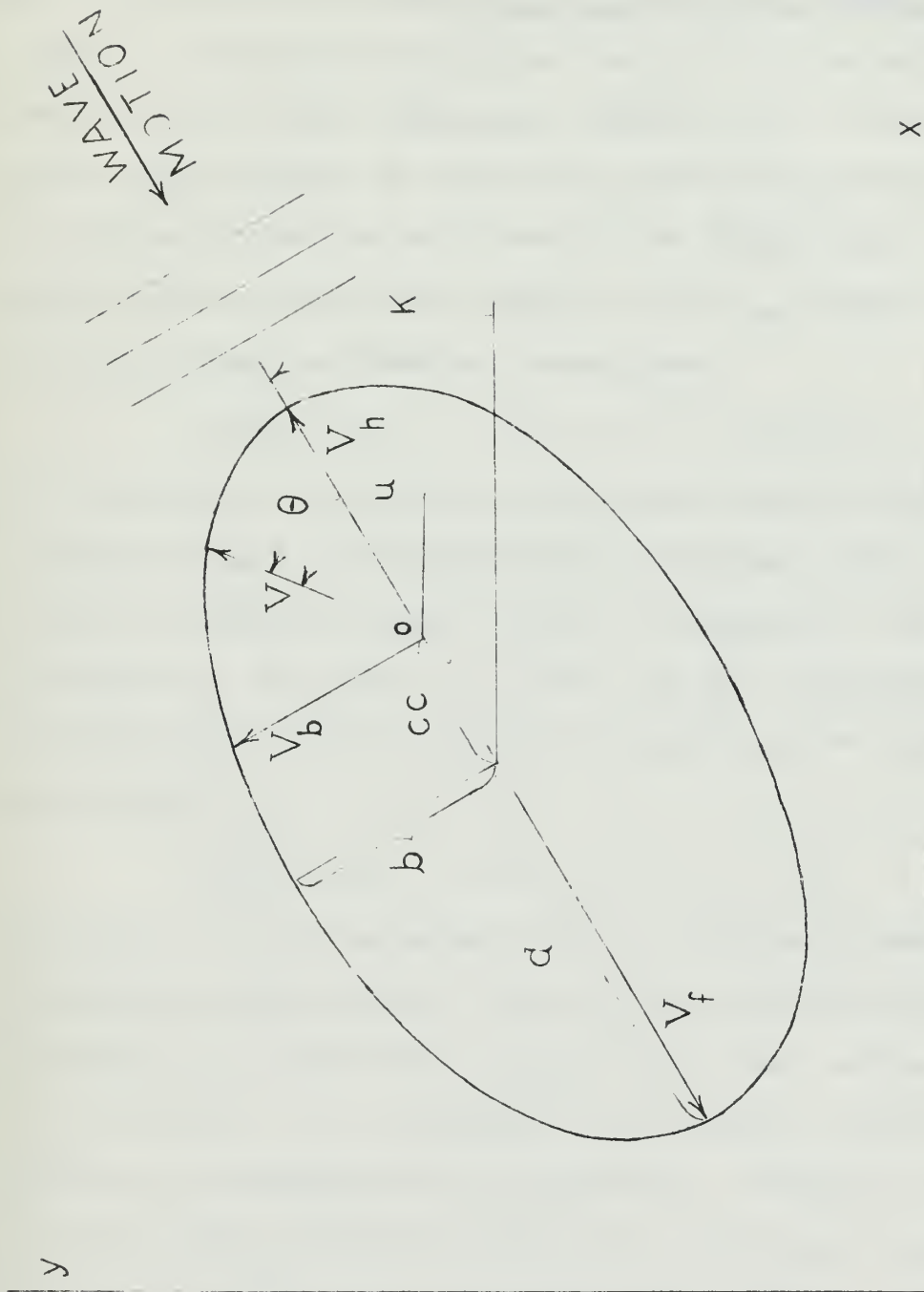


Figure 1. Polar Velocity Diagram



where the  $(\cdot)$  over a variable indicates its time derivative, and  $V = |\vec{V}| = V(x,y,u,t)$ . Hence, our problem is to choose the control angle,  $u(t)$ , at each point of the transit so as to minimize the cost function  $g$ .

## II. Abstract of the Theory

Consider a system of differential equations of the form  $\dot{x}^i = f^i$  ( $i=1, \dots, n$ ), where  $f^i = f^i(X, U, t)$  are functions of class  $C^2$ ,  $X$  is an  $n$ -dimensional vector,  $U$  is an  $m$ -dimensional vector, and  $t$ , the independent variable, is a scalar. We desire the solution to the above system of equations which is optimal in the sense of maximizing (minimizing) the terminal value of some cost function  $g(X, t)_{t=T}$ , while simultaneously satisfying terminal constraints,

$$h_i(X, t)_{t=T} = 0 \quad (i=1, \dots, N; \quad 0 \leq N \leq n).$$

The solution of a problem of the above form by the method of steepest ascent is concerned with the development of the  $m$  control variables,  $u_1(t), \dots, u_m(t)$ . Essentially the problem is solved in two parts. In part I we are interested in attaining admissibility, that is satisfying the terminal constraints

$$h_i(X, t)_{t=T} = 0 \quad (i=1, \dots, N; \quad 0 \leq N \leq n).$$

In part II we are concerned with the problem of maximizing (minimizing) the terminal value of the cost function  $g$ . The procedure is as follows: We initially guess nominal values of the control,  $\vec{U}$ ; in general, the resulting solution will neither be admissible nor an extremal. Changes in the control are then generated which tend to drive us to admissibility, that is,  $\vec{U}(t)$  is replaced by  $\vec{U}(t) + \delta \vec{U}(t)$ . This procedure is continued until we attain admissibility. The admissible solution, in general, will not be an extremal; so

we once again generate changes in the control which tend to drive us toward an extremal without affecting admissibility.

The author realizes the above is rather abbreviated; however, this discussion is intended to give a brief idea of what is to follow.

### III. Development of the Theory

#### Notation

Let us start by considering the following system of ordinary nonlinear differential equations:

$$\begin{aligned} \dot{x}^1 &= \dot{x} = V \cos u = f^1(x^1, x^2, u, t) \\ (3.1) \quad \dot{x}^2 &= \dot{y} = V \sin u = f^2(x^1, x^2, u, t) \\ \dot{x}^3 &= \dot{z} = g(x, y, t) = f^3(x^1, x^2, u, t) \end{aligned}$$

where  $V = V(x, y, u, t)$ ,  $u = u(t)$ , and  $g$  is the cost function to be minimized. We can write this as

$\dot{x}^i = f^i(x^1(t), x^2(t), u(t), t)$ , ( $i = 1, 2, 3$ ),  $0 \leq t \leq T$ , where  $t$ , the independent variable, is a scalar. It will be convenient in the development to follow to represent these sets of variables by matrices or by vectors, so let us define:

$$\vec{X} = \begin{bmatrix} x^1(t) \\ x^2(t) \\ x^3(t) \end{bmatrix} ; \vec{F} = \begin{bmatrix} f^1(x^1, x^2, u, t) \\ f^2(x^1, x^2, u, t) \\ f^3(x^1, x^2, u, t) \end{bmatrix} ; \vec{U} = [u(t)]$$

where  $\vec{U}$  is a single-rowed column matrix since there is but one control variable.

Equations (3.1) then become

$$(3.2) \quad \dot{\vec{X}} = \vec{F} .$$

The vector  $\vec{X}$  is called the state vector, or state, and the one dimensional vector  $\vec{U}$ , the control vector, or control. We will assume the control to be a continuous function of time.

Now the problem as posed in section I can be interpreted as follows: We desire to find a curve  $C$ , a solution to (3.2) for some choice  $\vec{U}$ , which is optimal in the sense of minimizing the terminal value of the cost function  $g$  while simultaneously satisfying the terminal constraints

$$x^1(T) = x_f^1, \quad x^2(T) = x_f^2, \quad ,$$

where  $x_f^1$  and  $x_f^2$  are the final values of the state corresponding to the rectangular coordinates of the terminal point in the OXY plane.

#### The Adjoint System of Equations

Of primary interest is the relationship between variations,  $\delta u(t)$ , in the control at any time,  $t$  ( $0 \leq t \leq T$ ), and the resulting terminal variations of the state variable,  $\delta \vec{X}(T)$ . If we consider (3.2) in component form, the variational relationships for the state variables can be expressed as:

$$\begin{aligned} \delta \dot{x}^1 &= \frac{\partial f^1}{\partial x^1} \delta x^1 + \frac{\partial f^1}{\partial x^2} \delta x^2 + \frac{\partial f^1}{\partial x^3} \delta x^3 + \frac{\partial f^1}{\partial u} \delta u \\ (3.3) \quad \delta \dot{x}^2 &= \frac{\partial f^2}{\partial x^1} \delta x^1 + \frac{\partial f^2}{\partial x^2} \delta x^2 + \frac{\partial f^2}{\partial x^3} \delta x^3 + \frac{\partial f^2}{\partial u} \delta u \\ \delta \dot{x}^3 &= \frac{\partial f^3}{\partial x^1} \delta x^1 + \frac{\partial f^3}{\partial x^2} \delta x^2 + \frac{\partial f^3}{\partial x^3} \delta x^3 + \frac{\partial f^3}{\partial u} \delta u \end{aligned}$$

Writing this in matrix form we have:

$$\delta \dot{\vec{X}} = G \delta \vec{X} + H \delta \vec{U}$$



where

$$G = \begin{bmatrix} \frac{\partial f^1}{\partial x^1} & \frac{\partial f^1}{\partial x^2} & \frac{\partial f^1}{\partial x^3} \\ \frac{\partial f^2}{\partial x^1} & \frac{\partial f^2}{\partial x^2} & \frac{\partial f^2}{\partial x^3} \\ \frac{\partial f^3}{\partial x^1} & \frac{\partial f^3}{\partial x^2} & \frac{\partial f^3}{\partial x^3} \end{bmatrix}; H = \begin{bmatrix} \frac{\partial f^1}{\partial u} \\ \frac{\partial f^2}{\partial u} \\ \frac{\partial f^3}{\partial u} \end{bmatrix}; \delta \dot{\vec{X}} = \begin{bmatrix} \delta \dot{x}^1 \\ \delta \dot{x}^2 \\ \delta \dot{x}^3 \end{bmatrix}$$

and  $\delta \vec{U} = [\delta u]$ .

Note in the special case where the control  $\vec{U}$  consists of a single variable,  $u$ ,  $\delta \vec{U}$  can be considered a single-rowed column vector.

If we now introduce a new set of variables

$$\Lambda(t) = \begin{bmatrix} \lambda(t) \\ \mu(t) \\ \sigma(t) \end{bmatrix},$$

Lagrange multipliers, which are undefined functions of  $t$ , and multiply (3.4) through by  $\Lambda^T$ , where  $\Lambda^T$  is the transpose of  $\Lambda$ , we obtain the scalar relationship

$$(3.5) \quad \Lambda^T \delta \dot{\vec{X}} = \Lambda^T G \delta \vec{X} + \Lambda^T H \delta \vec{U}.$$

Let us now integrate (3.5) over the interval  $(0, T)$  to obtain

$$(3.6) \quad \int_0^T \Lambda^T \delta \dot{\vec{X}} dt = \int_0^T \Lambda^T G \delta \vec{X} dt + \int_0^T \Lambda^T H \delta \vec{U} dt.$$

Integrating the integral on the left by parts yields

$$\Lambda^T \delta \vec{X} \Big|_0^T - \int_0^T \dot{\Lambda}^T \delta \vec{X} dt = \int_0^T \Lambda^T G \delta \vec{X} dt + \int_0^T \Lambda^T H \delta \vec{U} dt.$$

Rewriting this as

$$\Lambda^T \delta \vec{X} \Big|_0^T - \int_0^T (\dot{\Lambda}^T + \Lambda^T G) \delta \vec{X} dt = \int_0^T \Lambda^T H \delta \vec{U} dt$$

and choosing  $\Lambda$  as a solution to  $(\dot{\Lambda}^T + \Lambda^T G) \delta \vec{X} = 0$  defines the system of equations adjoint to the variational equations (3.4). That is, the adjoint system of equations is

$$(3.7) \quad \dot{\Lambda}^T = -\Lambda^T G \quad .$$

With this choice of  $\Lambda$ , (3.6) reduces to

$$(3.8) \quad \Lambda^T \delta \vec{X}|_T = \Lambda^T \delta \vec{X}|_0 + \int_0^T \Lambda^T H \delta \vec{U} \, dt$$

and, since we are assuming  $\delta \vec{X}(0) = 0$ , we have a relation between the terminal variations of the state variable,  $\delta \vec{X}(T)$ , and the variations in the control,  $\delta \vec{U}$ . Now if  $\Lambda$  is chosen as the specific solution to the adjoint system of equations, (3.7), such that at time  $t = T$ ,  $\Lambda(T) = (1, 0, 0)$ , then (3.8) becomes

$$\delta x^1(T) = \int_0^T \Lambda^T H \delta \vec{U} \, dt$$

giving the terminal variation of  $x^1$  due to variations  $\delta \vec{U}(t)$ . Similarly, choosing  $\Lambda_i(T)$  such that the  $i^{\text{th}}$  component is 1 and all other components are 0 yields

$$(3.9) \quad \delta x^i(T) = \int_0^T \Lambda_i^T H \delta \vec{U} \, dt$$

We may consider the matrix product  $\Lambda_i^T H$  as a vector, say  $\vec{A}_i(t)$ , when there is more than one control variable. Then (3.9) can be written as

$$\delta x^i(T) = \int_0^T \vec{A}_i(t) \cdot \delta \vec{U}(t) \, dt$$

and for a maximum change in  $x^i(T)$  we would choose  $\delta \vec{U}(t)$  parallel to  $\vec{A}_i(t)$ , for  $0 \leq t \leq T$ . Courant [3] page 223, refers to the vector  $\vec{A}_i(t)$  as the gradient or  $x^i$ , of  $\vec{A}_i = \vec{\nabla} x^i$ .

### Special Variations

From the previous discussion we have seen for a maximum change in  $x^i$  we would like  $\delta \vec{U}(t)$  to be parallel to  $\vec{V}x^i$  for  $0 \leq t \leq T$ . This motivates the following choice of a special variation of the control,

$$(3.10) \quad \delta \vec{U}(t) = e_1 \vec{V}x^1(t) + e_2 \vec{V}x^2(t) .$$

It might be worth noting we do not consider  $\vec{V}x^3$  since at this juncture we are only interested in satisfying the terminal constraints

$$x^1(t) = x_f^1, \quad x^2(t) = x_f^2 .$$

Substituting (3.10) into (3.9), and recalling  $\Lambda_i^T H = \vec{V}x^i$ , we obtain

$$(3.11) \quad \begin{aligned} \delta x^1(T) &= e_1 \int_0^T \vec{V}x^1 \cdot \vec{V}x^1 dt + e_2 \int_0^T \vec{V}x^1 \cdot \vec{V}x^2 dt \\ \delta x^2(T) &= e_1 \int_0^T \vec{V}x^1 \cdot \vec{V}x^2 dt + e_2 \int_0^T \vec{V}x^2 \cdot \vec{V}x^2 dt \end{aligned}$$

We now define

$$z^{ij} = \int_0^T \vec{V}x^i \cdot \vec{V}x^j dt$$

so that (3.11) becomes

$$\delta x^i(T) = \sum_{j=1}^2 e_j z^{ij} \quad (i=1,2) .$$

For small perturbations, the value of  $T$  will be changed only a small amount  $dT$ , so that

$$dx^i(T) = \delta x^i(T) + \dot{x}^i(T) dT \quad (i=1,2) .$$

We can express this relationship in matrix form as

$$(3.12) \quad \begin{bmatrix} z^{11} & z^{12} & \dot{x}^1 \\ z^{21} & z^{22} & \dot{x}^2 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ dT \end{bmatrix} = \begin{bmatrix} x_f^1 - x^1(T) \\ x_f^2 - x^2(T) \end{bmatrix}$$

or  $ZE = \Delta \vec{X}$ . Let us now digress for a moment, and recall that if we have an overdetermined system of equations, say  $AX = B$ , where  $A$ , the coefficient matrix, is  $3 \times 2$ ,  $X$ , the unknown matrix, is  $2 \times 1$ , and  $B$  is a  $3 \times 1$  matrix, then a least squares solution for  $X$  is obtained by multiplying both sides of the above equation on the left by  $A^T$  provided  $AA^T$  is not singular. Since we have a system of two equations in three unknowns, let us apply the above technique in reverse.

$$\text{Let } E = Z^T C \text{ where } C = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \text{ so that (3.12)}$$

becomes  $ZZ^T C = \Delta X$ . Now  $ZZ^T$  is a  $2 \times 2$  Gram matrix, so let

$$ZZ^T = A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}.$$

Thus (3.12) is representable as  $AC = \Delta X$ , and we are now able to solve for  $C$ . Use of (3.10) and  $E = Z^T C$  gives us our variations in control  $\delta \vec{U}(t)$ , for  $0 \leq t \leq T$ .

At this point in the development we should be able to obtain an admissible curve. We will now consider the problem of decreasing the cost function without affecting  $x^1(T)$  and  $x^2(T)$ . To accomplish this end we consider a special variation of the form

$$(3.13) \quad \delta \vec{U}(t) = \alpha_1 \vec{v}_x^1(t) + \alpha_2 \vec{v}_x^2(t) + \alpha_3 \vec{v}_x^3(t) \quad .$$

Substituting (3.13) into (3.9) yields

$$\begin{aligned} \delta x^i(T) &= \sum_{j=1}^3 \alpha_j z^{ij} \quad \text{and} \\ dx^i(T) &= \delta x^i(T) + \dot{x}^i(T) dT \end{aligned} \quad (i=1,2,3) \quad .$$

In matrix notation this becomes

$$(3.14) \quad \begin{bmatrix} x_f^1 - x^1(T) \\ x_f^2 - x^2(T) \\ x_f^3 - x^3(T) \end{bmatrix} = \begin{bmatrix} z^{11} & z^{12} & z^{13} & \dot{x}^1(T) \\ z^{21} & z^{22} & z^{23} & \dot{x}^2(T) \\ z^{31} & z^{32} & z^{33} & \dot{x}^3(T) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ dT \end{bmatrix}$$

Since we have an admissible path,

$$x_f^1 - x^1(T) \approx x_f^2 - x^2(T) \approx 0 ;$$

however,  $x_f^3 - x^3(T)$  is unknown, and, in general, is not zero.

Once again we are faced with the problem of solving an under-determined system of equations, (3.14). As before we will

let  $\alpha = Z^T D$  where  $D = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$  so that (3.14) becomes  $\Delta \vec{X} = Z Z^T D$ .

Now  $Z Z^T$  is a 3 x 3 symmetric Gram matrix so let

$$Z Z^T = B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{12} & b_{22} & b_{23} \\ b_{13} & b_{23} & b_{33} \end{bmatrix}$$

Thus (3.14) is representable as

$$(3.15) \quad \Delta \vec{X} = B D$$



Define  $d_i = \varepsilon \beta_{i3}$  ( $i=1,2,3$ ), where  $\varepsilon$  is to be determined, and  $\beta_{i3}$  as the cofactor of  $b_{i3}$  in the determinant of B, that is

$$\begin{aligned} d_1 &= \varepsilon [b_{12} b_{23} - b_{22} b_{13}] \\ d_2 &= \varepsilon [b_{12} b_{13} - b_{11} b_{23}] \\ d_3 &= \varepsilon [b_{11} b_{22} - b_{12} b_{12}] . \end{aligned}$$

Then, upon substitution into (3.15), we obtain

$$(3.16) \quad \Delta X = \begin{bmatrix} x_f^1 - x^1(T) \\ x_f^2 - x^2(T) \\ x_f^3 - x^3(T) \end{bmatrix} = \varepsilon \begin{bmatrix} 0 \\ 0 \\ \det B \end{bmatrix} .$$

Thus, by choosing  $\varepsilon$  negative, we can decrease  $x^3$  and keep  $x^1$  and  $x^2$  fixed so long as B is nonsingular. That is

$$\frac{dx^1}{d\varepsilon} = \frac{dx^2}{d\varepsilon} = 0 \text{ and } \frac{dx^3}{d\varepsilon} = \det B.$$

Now, having D and  $\alpha = Z^T D$ , (3.13) may be solved for  $\delta \vec{U}$  as a multiple of  $\varepsilon$ . In order to limit the maximum change in the control, we define the following norm:

$$N = \left\{ \max_t |u(t)| + 1 \left| \frac{dT}{T} \right| \right\} .$$

We now define  $\varepsilon = -\frac{DELUMX}{N}$ , where DELUMX is the maximum allowable change in the control. Thus  $\delta \vec{U}(t)$ , the solution to (3.13) is replaced by

$$\varepsilon \delta \vec{U}(t) = -\frac{DELUMX}{N} \delta \vec{U}(t) .$$

#### IV. The Numerical Solution

The numerical solution of the problem as posed in Section I is based upon the solution of the original system of equations, the adjoint equations, and the variational equations. For the sake of simplicity, a finite difference approximation to the derivative with a variable time step of integration is used to integrate the above equations. The variable time step of integration is obtained by dividing the total time as determined after each path into two hundred equal increments. Initially, the value for  $T$ , the terminal time, is determined by the geodesic track, but this is not a requirement. Any reasonable guess for  $T$  will suffice. Justification for this method of integration is based upon a comparison of the geodesic route as obtained, using the above mentioned technique and a fourth order Runge-Kutta integration method.

Due to the core storage limitation imposed by the Fleet Numerical Weather Central, 165,000 octal words, a theatre of operation is established which is a subfield of the  $63 \times 63$  grid mentioned in Section I. Having established this subfield the wave field data is then read into core memory. We are now in a position to proceed with the numerical solution of the problem.

Although not a requirement, the geodesic track from the point of departure to the point of destination is calculated for comparison purposes. As stated in Section II, a nominal

control program is initially required. This is taken as the angle of inclination of the straight line connecting the track end points.

Recall that in the development of the theory we chose as particular solutions to the adjoint equation,  $\Lambda_1$ ,  $\Lambda_2$ , and  $\Lambda_3$ , where  $\Lambda_i(T)$  has its  $i^{\text{th}}$  component equal to 1 and all other components equal to 0. To satisfy this requirement we define

$$\Lambda_1(0) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \Lambda_2(0) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \Lambda_3(0) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

and integrate the adjoint system of equations forward to obtain  $\Lambda_1(T)$ ,  $\Lambda_2(T)$ ,  $\Lambda_3(T)$ . The resulting matrix

$$\begin{bmatrix} \lambda_1(T) & \lambda_2(T) & \lambda_3(T) \\ \mu_1(T) & \mu_2(T) & \mu_3(T) \\ \sigma_1(T) & \sigma_2(T) & \sigma_3(T) \end{bmatrix}$$

is then inverted to obtain new initial values  $\Lambda_1(0)$ ,  $\Lambda_2(0)$ ,  $\Lambda_3(0)$  so that upon integration forward a second time one obtains

$$\Lambda_1(T) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \Lambda_2(T) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \text{ and } \Lambda_3(T) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Having obtained the initial values of the adjoint vectors  $\Lambda_1$ ,  $\Lambda_2$ , and  $\Lambda_3$  such that at the terminal time  $T$  they take on prescribed values, we will run a path, check for admissibility, and if not admissible generate corrections to the control. Simultaneous integration of the original system of equations and the adjoint equations gives a path. At each point along this path the gradients in the  $x^i$  direction ( $i=1,2,3$ ) are

calculated and stored for later recall in determining the changes in the control. In addition, the elements of the Z matrix of Section III are calculated. A check is now made to see if we have attained admissibility. If not, we proceed to calculate the changes in control which will tend to satisfy the admissibility constraint. Assuming we do not have an admissible path, the A, C, and E matrices of Section III are calculated. Thus we obtain new values for the control and the terminal time T, that is,  $\vec{U}(t)$  is replaced by  $\vec{U}(t) + \delta\vec{U}(t)$  and T is replaced by  $T + dT$ . The above procedure is continued until the admissibility constraint is met.

Having admissibility we consider the problem of decreasing the cost function

$$g = \int_0^T e^{\gamma} H dt ,$$

where  $\gamma$  is a scalar, without affecting admissibility. A check is made to see if the terminal value of the cost function as determined by the last admissible path is less than that determined by the previous admissible path. If it is, we proceed to calculate the changes in control as defined in Section III, so as to decrease the cost function  $g$ . If not, we decrease the maximum allowable change in the control at any time  $t$  by one-half. A check is then made to determine if the maximum allowable change in control is less than 0.01. If it is, a solution has been obtained. If not, proceed to calculate the changes in control so as to decrease the cost function. Assuming we have no solution we calculate the B, D, and  $\alpha$  matrices of Section III. This enables one to obtain



new values of the control, that is,  $\vec{U}(t)$  is replaced by  $\vec{U}(t) + \delta\vec{U}(t)$ . A new path is run which, in general, is not admissible; so we again drive to admissibility. This procedure is continued until a solution is obtained.

It should be noted the values used in the program for the maximum allowable change in the control and the stopping criterion are quite arbitrary.



## V. Conclusions

The value of this study lies not in the solution of the particular problem considered, but in the fact that given any realistic cost function which is a function of the stereographic grid coordinates  $x$  and  $y$ , the control  $U$ , and time, one can obtain an optimum solution with only slight modifications. Such a cost function for a cargo type vessel might reflect costs due to storm damage, failure to meet scheduled commitment dates, and spoilage of cargo due to extended time at sea, to mention but a few.

On the basis of the results obtained the following observations seem justified:

- (a) It was shown in Section III that a necessary condition for the solution curve  $C$  to be an extremal is that the determinant of the  $B$  matrix at  $t = T$  approach zero as track iteration continues. In the case where we chose  $\gamma = 0$ , that is, minimum time, we saw that this, in fact, was the case. In the second problem, where we chose  $\gamma = 0.01$ , the determinant of the  $B$  matrix was reduced from a value on the order of 5,000 to 0.9. The conclusion to be drawn here is that, although the solution curve  $C$  is not an extremal, it very closely approximates one.
- (b) Defining the initial control as the constant function  $u(t) = V$  ( $0 \leq t \leq T$ ), where  $V$  is the initial angle of departure, leads to convergence to within fifteen miles of the point of destination.

Thus there seems to be no problem as to how to initially choose the control so as to obtain convergence.

(c) The simplified integration routine used is feasible and yields satisfactory results.

(d) The requirement of storing an entire control program and a complete set of functions  $\Lambda_i^T H$   $i = 1, 2, 3$  for any one iteration is a marked disadvantage. Due to the large storage requirements imposed by the establishment of the wave fields a relatively small amount of core memory remains.

(e) An analysis of the results obtained show a substantial decrease in the wave heights for the minimum time route, as compared to the geodesic track, with a decrease in the total transit time of approximately  $2\frac{1}{2}$  hours. The weighted time route, on the other hand, shows only slight decreases in the wave heights encountered, as compared to the minimal time track. This leads one to conclude that if the objective is to limit the wave heights encountered, a function other than  $\exp \gamma H$  should be used. This is more readily seen if we expand  $e^{\gamma H}$ , that is,

$$e^{\gamma H} = 1 + \gamma H + \frac{(\gamma H)^2}{2!} + \frac{(\gamma H)^3}{3!} + \dots$$

For small values of  $\gamma$  and nominal wave heights,  $H$ , the dominant factor in minimizing  $z = \int_0^T e^{\gamma H} dt$  is time.

# BIBLIOGRAPHY

1. Bryson, A. E., and W. F. Denham. A Steepest Ascent Method for Solving Optimum Programming Problems, Journal of Applied Mechanics, June 1962, pp. 247-257.
2. Bleick, W. E., A General Ship Routing Program, NPS-53BL8051B, 1 May 1968.
3. Courant, R., and D. Hilbert, Methods of Mathematical Physics, Vol. 1, Interscience Publishers, Inc., New York, 1953.
4. Haltiner, G. J., W. E. Bleick, and F. D. Faulkner, Use of Long-Range Weather Forecasts in Ship Routing, Naval Postgraduate School Technical Report/Research Paper No. 78, 1 July 1967.
5. Bleick, W. E., and F. D. Faulkner, Minimal-Time Ship Routing, Journal of Applied Meteor., Vol. 4, No. 2, April 1965, pp. 217-221.
6. James, R. W., Application of Wave Forecasts to Marine Navigation, U. S. Navy Hydrographic Office, 1959.

## APPENDIX

### COMPUTER PROGRAM

```

PROGRAM TEST (INPUT,OUTPUT,TAPE4)
  DIMENSION DTG(12),U(200),DELTA(200),GRADX(200),GRADY(200),
  1 TFS(201),HT(201),AZ(201),AL(201),WA(201),WE(39H9),WD(63,63)
  2 XLAM(3),TIAM(3),XMU(3),XSIGMA(3),GRADZ(200),TMU(2)
  COMMON Y,Y,A,R,CC,H,CK,SK,KX,KY,KT,FOX,FOY,KCON
  COMMON/L1/XHT(8448),CSK(8448),SNK(8448),T
  COMMON/L2/HX,HY
  COMMON/L5/CAPV,CAPVX,CAPVY,CAPVU,W
  COMMON/L6/XK,XLG,XLT
  EQUIVALENCE (WE(1),WD(1,1))
  REAL MSF
  INTEGER DTG
  WE WILL FIRST ESTABLISH THE WAVE FIELD IN OUR THEATER OF INTEREST
  ITAX=0 & DTGX=0 & IMINX=0 & JMINX=0 & KTX=0
  READ DIMENSIONS AND MINIMUM INDICES OF ARRAYS
  READ 5,KX,KY,KT,IMIN,JMIN,CRIT
  5 FORMAT (5I2,F4.0)
  PRINT 3,KX,KY,KT,IMIN,JMIN,CRIT
  3 FORMAT(33H0DIMENSION OF WAVE FIELD ARRAYS=(I3,1H,I3,1H,I3,1H)4X5HT
  IMIN=I3,4X5HJMIN=I3,4X5HCRIT=F4.0)
  IOX=32-IMIN
  JOY=32-JMIN
  FOX=IOX
  FOY=JOY
  KCON=KX*KY+KX
  XMEAN=FLOAT(KX+1)/2.
  YMEAN=FLOAT(KY+1)/2.
  FOX=XMEAN-2.01
  FOY=YMEAN-2.01
  CLEAR DTG ARRAY
  DO 6 I=1,KT
  6 DTG(I)=0
  READ SOCIAL DATE-TIME GROUPS OF THE TIME SERIES MEMBERS
  READ 11,DTG(1),ITAU
  11 FORMAT(020,8X,I2)
  IF (DTGX.EQ.DTG(1).AND.ITAX.EQ.ITAU.AND.IMINX.EQ.IMIN.AND.
  1JMINX.EQ.JMIN.AND.KTX.EQ.KT)750,625
  SET ARRAY DTG TO SELECTED TIME SERIES
  625 JJ=KT-1
  DO 7 I=1,JJ
  7 CALL DTGK (DTG(I),ITAU,DTG(I+1))
  7 CONTINUE
  PROCESS THE UNPACKED ARRAYS
  REWIND 4
  DO 700 K=1,KT
  KKK=((K-1)*KY-1)*KX
  DO 650 KK=1,2
  NG=0
  KW=KK-1
  CALL FIELD SX(WE,DTG(K),KW,NG)
  IF (NG-1)90,89,90
  89 PRINT 91,DTG(K)
  91 FORMAT(14H0FIELD OF DTG=020,1X31HNOT FOUND ON INPUT TAPE4. ABORT/)
  GO TO 83
  90 GO TO (92,94) KK
  92 DO 8 I=1,KX
  II=KKK+I

```



```

      DELX=J-TOX
      DO 8 J=1,KY
      JJ=KX*J+II
      DFLY=J-JOY
      ROOT=SQRT( DELX*DELX + DELY*DELY )
      ARG=WD(I+IMIN,J+JMIN)/5.729577951
      COS=COSF(ARG)
      SIN=SINF(ARG)
      CSK(JJ)=(-DELX*COS-DFLY*SIN)/ROOT
8     SNK(JJ)=(DELX*SIN-DFLY*COS)/ROOT
      GO TO 650
94    DO 10 I=1,KX
      II= KKK + I
      DO 10 J=1,KY
      JJ = KX*J + II
      10 XHT(JJ)=WD(I+IMIN,J+JMIN)
650   CONTINUE
      IF(SENSE SWITCH 1)610,700
610   PRINT 1100. K,DTG(K)
1100  FORMAT(1H0,J2.2H. ,020)
      DO 620 J=1,KY
620   PRINT 1200. ( WD( I+IMIN,J+JMIN ),I=1,KX )
1200  FORMAT( 1X,26F5.1/ )
700   CONTINUE
      REWIND 4
      DTGX=DTG(1) $ ITAUX=ITAU $ IMINX=IMIN $ JMINX=JMIN $ KTX=KT
      DO 750 I=1,KX
      DO 750 J=1,KY
      JJ=KX*J + I
      RI=I+IMIN $ RJ=J+JMIN $ KER=0
      CALL SEALAND (RI,RJ,LAND,KER)
      IF(KER)646,648,646
646   PRINT 647 $ GO TO 83
647   FORMAT(26H0RI,RJ INDEX ERRORS, ABORT/)
648   IF(LAND)649,750,649
649   DO 749 K=1,KT
      KKK=((K-1)*KY-1)*KX + JJ
749   XHT(KKK)=20.0
750   CONTINUE
C WE SHOULD NOW HAVE OUR WAVE FIELD DATA STORED IN MEMMORY
C READ HOUR OF DEPARTURE AND COORDINATES OF TRACK END POINTS
      READ 13,HR,XLG1,XLT1,XLG2,XLT2
      13 FORMAT(F3.0,F6.1,F5.1,F6.1,F5.1)
      PRINT 14,HR,XLG1,XLT1,XLG2,XLT2
      14 FORMAT(21H ROUTE OF SHIP BEGINSF4.0,31H HOURS AFTER TIME SERIES OR
      1IGIN/19H FROM LONGITUDE = F6.1,16H AND LATITUDE = F6.1,19H TO
      2 LONGITUDE = F6.1,16H AND LATITUDE = F6.1)
C WE WILL NOW CONVERT COORDINATES OF TRACK END POINTS TO GRID VALUES
      ARG=(XLG1-10.)/57.29577951
      COSLG1=COSF(ARG)
      SINLG1=SINF(ARG)
      ARG=(XLT1-10.)/57.29577951
      COSLT1=COSF(ARG)
      SINLT1=SINF(ARG)
      ARG=XLG2/57.29577951
      COSLG2=COSF(ARG)
      SINLG2=SINF(ARG)
      ARG=XLT2/57.29577951

```

```

COSLT2=COSF (ARG)
SINLT2=SINF (ARG)
PR1=31.205*COSLT1/(1.+SINLT1)
X1=PR1*COSLG1
Y1=PR1*SINLG1
PR2=31.205*COSLT2/(1.+SINLT2)
X2=PR2*COSLG2
Y2=PR2*SINLG2
XSTART=X1+FOX
YSTART=Y1+FOY
XEND=X2+FOX
YEND=Y2+FOY
C COMPUTE GEODESIC ROUTE
EL=SINLT2*COSLT1*SINLG1 - COSLT2*SINLT1*SINLG2
EM= - SINLT2*COSLT1*COSLG1 + COSLT2*SINLT1*COSLG2
EN= (SINLG2*COSLG1 - COSLG2*SINLG1)*COSLT1*COSLT2
ROOT=SQRTF (EL*EL + EM*EM + EN*EN)
EL=EL/ROOT
EM=EM/ROOT
EN=EN/ROOT
DELX=X2-X1
DELY=Y2-Y1
S12=SQRTF (DELX*DELX + DELY*DELY)
ARC=S12
IF (XLG2-XLG1)20,21,20
20 ARG=ABSF (EN/62.41)
ARC=ASINF (ARG*S12)/ARG
21 X=XSTART
Y=YSTART
T=HR/24.
XTFS=0.0
CALL TFRP
HT(1)=H
IFS(1)=XTFS
CALL ANGLEF
AZ(1)=XLG
AL(1)=XLT
KA(1)=XK
KK=1
XX=XSTART
YY=YSTART
SS=0.0
LR=0
STEP=1./6.
DO 40 K=2,200
COSP=(FOY-Y)*EN/31.205 + EM
SINP=(X-FOX)*EN/31.205 - EL
CALL TFRP
CALL SHIP
CALL VDEFIV
X=XX + CAPV*COSP*STEP
Y=YY + CAPV*SINP*STEP
S=SS + CAPV*STEP
IF (S-ARC)35,34,34
34 RAT=(ARC-S)/(S-SS)
T=RAT*STEP + T
XTFS=RAT*STEP + XTFS
X=(X-XX)*RAT + XX

```

```

      Y=(Y-YY)*PA1 + YY
      GO TO 15
35  XX=X
      YY=Y
      SS=S
      T=T + STEP
      XTFS=XIFS + STEP
      IF( (K-1)*2/6 + 1 - NK)16,16,15
15  NK=NK + 1
      CALL TERP
      HT(NK)=H
      TFS(NK)=XTFS
      CALL ANGLE
      AZ(NK)=XLG
      AL(NK)=XLT
      WA(NK)=XK
      IF(S-ARC)16,41,41
16  IF(ABS(X-XMEAN)-POX)37,524,524
37  IF(ABS(Y-YMEAN)-POY)40,524,524
40  CONTINUE
41  PRINT 17
17  FORMAT(////30X14HGEODESIC ROUTE)
      PRINT 18
18  FORMAT( 5X4HDAY56X5HLONGI4X5HIATI-5X4HWAVE5X14HWAVE DIRECTION/2X9H
10F TRAVEL4X5H-TUDE4X4HTUDE5X4HHEIGHT6X10HFROM NORTH/)
      PRINT 19,(TFS(K),AZ(K),AL(K),HT(K),WA(K),K=1,NK)
19  FORMAT ( F9.2,F11.1,F8.1,F10.2,F13.0 )
C WE WILL NOW GUESS AN INITIAL VALUE OF THE CONTROL U(T) AND RUN A PATH.
      DELTAX=XEND-XSTART
      DELTAY=YEND-YSTART
      IF(DELTAX.EQ.0.0)GO TO 530
      ARG=DELTAY/DELTAX
      ALFA=ATANF(ARG)
      IF(DELTAX)531,530,532
532 IF(DELTAY)533,534,534
533 V=6.28319 + ALFA
      GO TO 535
534 V=ALFA
      GO TO 535
531 V=3.14159 + ALFA
      GO TO 535
530 IF(DELTAY)536,537,538
536 V=4.7124
      GO TO 535
537 V=1.5708
      GO TO 535
537 PRINT 539
539 FORMAT(5X4HSTOP)
      GO TO 83
535 CONTINUE
      NN=0
      GAMA=0.0
541 LR=1
      RKX=FLOATF(KX)-1.01
      RKY=FLOATF(KY)-1.01
      NN=0
      L=0
      DFLUMX=0.4

```

```

      TAUOLD=30.0
      ZOLD=10000.
      TAU=XTFS
      DO 500 I=1,200
500  U(I)=V
501  TLAM(1)=1.0
      XMU(1)=0.0
      XSIGMA(1)=0.0
      TLAM(2)=0.0
      XMU(2)=1.0
      XSIGMA(2)=0.0
      TLAM(3)=0.0
      XMU(3)=0.0
      XSIGMA(3)=1.0
      X=XSTART
      Y=YSTART
      STEP=TAU/200.
      T=HR/24.
      DO 502 I=1,200
      CALL TERP
      ARG=GAMA*H
      CALL SHIP
      W = U(I)
      CALL VDFRIV
      DO 550 J=1,3
      XLAM(J)=TLAM(J)-(CAPVX*COSF(U(I))*TLAM(J)+CAPVX*SINF(U(I))*XMU(J)
1  +GAMA*EXPF(ARG)*HX*XSIGMA(J))*STEP
      XMU(J)=XMU(J)-(CAPVY*COSF(U(I))*TLAM(J)+CAPVY*SINF(U(I))*XMU(J)
1  +GAMA*EXPF(ARG)*HY*XSIGMA(J))*STEP
550  TLAM(J)=XLAM(J)
      X=X+CAPV*COSF(U(I))*STEP
      Y=Y+CAPV*SINF(U(I))*STEP
      IF(X.LT.2.01)GO TO 524
      IF(Y.LT.2.01)GO TO 524
      IF(X.GT.RKX)GO TO 524
      IF(Y.GT.RKY)GO TO 524
      T=T+STEP
502  CONTINUE
      TMU(1)=XMU(1)
      TMU(2)=XMU(2)
      ADJDET= XLAM(1)*XMU(2)-XLAM(2)*XMU(1)
      TLAM(1)=XMU(2)/ADJDET
      XMU(1)=-XMU(1)/ADJDET
      XSIGMA(1)=0.0
      TLAM(2)=-XLAM(2)/ADJDET
      XMU(2)=XLAM(1)/ADJDET
      XSIGMA(2)=0.0
      TLAM(3)=(XLAM(2)*XMU(3)-XLAM(3)*TMU(2))/ADJDET
      XMU(3)=(XLAM(3)*TMU(1)-XLAM(1)*XMU(3))/ADJDET
      XSIGMA(3)=1.0
      X=XSTART
      Y=YSTART
      Z=0.0
      T=HR/24.
      XTFS=0.0
      Z11=0.0
      Z12=0.0
      Z13=0.0

```

```

Z22=0.0
Z23=0.0
Z33=0.0
DO 503 I=1,200
CALL TERP
ARG=GAMA*H
HT(I)=H
TFS(I)=XTFS
CALL ANGLE
A7(I)=XIG
AL(I)=XIT
WA(I)=XK
CALL SHIP
W = U(I)
CALL VDERIV
GRADX(I)=TLAM(1)*(CAPVU*COSF(U(I))-CAPV*SINF(U(I))) + XMU(1)*
1 (CAPVU*SINF(U(I)) + CAPV*COSF(U(I)))
GRADY(I)=TLAM(2)*(CAPVU*COSF(U(I))-CAPV*SINF(U(I))) + XMU(2)*
1 (CAPVU*SINF(U(I)) + CAPV*COSF(U(I)))
GRADZ(I)=TLAM(3)*(CAPVU*COSF(U(I))-CAPV*SINF(U(I))) + XMU(3)*
1 (CAPVU*SINF(U(I)) + CAPV*COSF(U(I)))
Z11=Z11+GRADX(I)*GRADX(I)*STEP
Z12=Z12+GRADX(I)*GRADY(I)*STEP
Z13=Z13+GRADX(I)*GRADZ(I)*STEP
Z22=Z22+GRADY(I)*GRADY(I)*STEP
Z23=Z23+GRADY(I)*GRADZ(I)*STEP
Z33=Z33+GRADZ(I)*GRADZ(I)*STEP
DO 551 J=1,3
XLAM(J)=TLAM(J)-(CAPVX*COSF(U(I))*TLAM(J)+CAPVX*SINF(U(I))*XMU(J)
1 +GAMA*FXPF(ARG)*HX*XSIGMA(J))*STEP
XMU(J)=XMU(J)-(CAPVY*COSF(U(I))*TLAM(J)+CAPVY*SINF(U(I))*XMU(J)
1 +GAMA*FXPF(ARG)*HY*XSIGMA(J))*STEP
551 TLAM(J)=XLAM(J)
X=X+CAPV*COSF(U(I))*STEP
Y=Y+CAPV*SINF(U(I))*STEP
Z=Z + EXPF(ARG)*STEP
I=T+STEP
XTFS=XTFS+STEP
503 CONTINUE
XDOT=CAPV*COSF(U(200))
YDOT=CAPV*SINF(U(200))
ARG=GAMA*H
ZDOT=EXPF(ARG)
CALL TERP
HT(201)=H
TFS(201)=XTFS
CALL ANGLE
A7(201)=XIG
AL(201)=XIT
WA(201)=XK
XDIFF=XFND-X
YDIFF=YFND-Y
DELX=X-FOX
DELY=Y-FOY
MSF=(DELX**2+DELY**2+973.75)/1043.638743
ERROR=XDIFF**2 + YDIFF**2
IF ( ERROR=MSF*MSF*CRIT*CRIT*0.00002366 ) 505,505,511
511 A11=Z11**2+Z12**2+XDOT**2

```



```

A12=Z11*Z12+Z22*Z12+XDOT*YDOT
A22=Z12**2+Z22**2+YDOT**2
DET=A11*A22-A12**2
C1=(A22*XDIFF-A12*YDIFF)/DET
C2=(A11*YDIFF-A12*XDIFF)/DET
E1=Z11*C1+Z12*C2
E2=Z12*C1+Z22*C2
DELTAU=C1*XDOT+C2*YDOT
TAU=TAU+DELTAU
DO 506 K=1,200
DELU(K)=E1*GRADX(K)+E2*GRADY(K)
506 U(K)=U(K)+DELU(K)
N=N+1
IF(N.GT. 25)GO TO 507
GO TO 501
507 PRINT 510
510 FORMAT(/5X14HNO CONVERGENCE)
GO TO A3
505 CONTINUE
L=L+1
IF(NN.EQ.1) GO TO 544
IF(TAU.LT. TAUOLD)GO TO 512
DELUXX=DELUXX/2.
IF(DELUXX.LT. 0.01) GO TO 514
512 TAUOLD=TAU
GO TO 546
544 IF(Z.LT.ZOLD) GO TO 545
DELUXX=DELUXX/2.
IF(DELUXX.LT.0.01) GO TO 514
545 ZOLD=Z
546 R11=Z11*Z11 + Z12*Z12 + Z13*Z13 + XDOT*XDOT
R12=Z12*Z11 + Z12*Z22 + Z23*Z13 + XDOT*YDOT
R13=Z13*Z11 + Z23*Z12 + Z33*Z13 + XDOT*ZDOT
R22=Z12*Z12 + Z22*Z22 + Z23*Z23 + YDOT*YDOT
R23=Z13*Z12 + Z23*Z22 + Z33*Z23 + YDOT*ZDOT
R33=Z13*Z13 + Z23*Z23 + Z33*Z33 + ZDOT*ZDOT
D=R11*(R22*R33-R23*R23)-R12*(R12*R33-R23*R13)
1 +R13*(R12*R23-R22*R13)
D1=(R13*R22-R12*R23)/D
D2=(R11*R23-R13*R12)/D
D3=(R12*R12-R11*R22)/D
R1=Z11*D1 + Z12*D2 + Z13*D3
R2=Z12*D1 + Z22*D2 + Z23*D3
R3=Z13*D1 + Z23*D2 + Z33*D3
R4=XDOT*D1 + YDOT*D2 + ZDOT*D3
BIG=0.0
DO 515 I=1,200
DELU(I)=R1*GRADX(I) + R2*GRADY(I) + R3*GRADZ(I)
RR=ABSF( DELU(I) )
IF( RR.GT. BIG )BIG=RR
515 CONTINUE
RR = DELUXX/( BIG +1.0*ABSF(R4)/TAU )
DO 520 I=1,200
DELU(I)=RR*DELU(I)
520 U(I)=U(I) + DELU(I)
IF(L.GT. 25)GO TO 521
GO TO 501
514 CONTINUE

```

```

      IF( NN .GT. 0 )GO TO 540
      PRINT 522
522  FORMAT(////28X18HMINIMUM TIME ROUTE)
      PRINT 543.0
543  FORMAT(//20X7HDET R =F6.2)
      PRINT 18
      PRINT 19. ( TFS(K),A7(K),AL(K),HT(K),WA(K),K=1,201,10 )
      GAMA=0.01
      NN=1
      GO TO 541
540  PRINT 542
542  FORMAT(////28X19HWEIGHTED TIME ROUTE)
      PRINT 543.0
      PRINT 18
      PRINT 19. ( TFS(K),A7(K),AL(K),HT(K),WA(K),K=1,201,10 )
      GO TO 83
521  PRINT 523
523  FORMAT(5X8HNO SOL N)
      GO TO 83
524  PRINT 525.X,Y
525  FORMAT(5X40HSHIP IS NOW OUTSIDE THEATER OF OPERATION5X3HX =F10.5,
1 5X3HY =F10.5)
83  STOP
      END

```

```

SUBROUTINE TERP
  DIMENSION C(4),P(4),Q(4),PX(4),QY(4),HP(4),CP(4),SP(4),HPX(4),
  1HPY(4),CPX(4),CPY(4),SPX(4),SPY(4),HD(4),CD(4),SD(4),HS(4),CS(4),
  2SS(4),HT(4,4),CT(4,4),ST(4,4)
  COMMON X,Y,A,R,CC,H,CK,SK,KX,KY,KT,FOX,FOY,KCON
  COMMON/L1/XHT(8448),CSK(8448),SNK(8448),T
  COMMON/L2/HX,HY
  COMMON/L3/DKX,DKY
  L=XFIXF(T)
  TT=(-INTF(T)+T)*2.-1.
  TP1=TT+1.
  TM1=TT-1.
  T2M=TP1*TM1
  IF(L-KT+3)1,1,4
1 IF(L)2,2,3
2 K4=3
  TM3=TT-3.
  C(1)=TM1*TM3/8.
  C(2)=-TP1*TM3/4.
  C(3)=T2M/8.
  GO TO 16
3 K4=4
  G=(3.*TT+2.)*TT-9.
  F=-4.*TT+G
  C(1)=-T2M*TM1/16.
  C(2)=G*TM1/16.
  C(3)=-F*TP1/16.
  C(4)=T2M*TP1/16.
  GO TO 15
4 IF(L-KT)6,5,5
5 K4=1
  L=KT-1
  C(1)=1.
  GO TO 16
6 G=(3.*TT+2.)*TT-9.
  F=-4.*TT+G
  C(1)=-T2M*TM1/16.
  IF(L-KT+1)11,10,5
10 K4=2
  C(2)=(G*TM1+(T2M-F)*TP1)/16.
  GO TO 15
11 C(2)=G*TM1/16.
  IF(L-KT+2)3,12,10
12 K4=3
  C(3)=(T2M-F)*TP1/16.
15 L=L-1
16 M=XFIXF(X)-2
  N=XFIXF(Y)-2
  XX=(-INTF(X)+X)*2.-1.
  YY=(-INTF(Y)+Y)*2.-1.
  XP1=XX+1.
  XM1=XX-1.
  YP1=YY+1.
  YM1=YY-1.
  X2M=XP1*XM1
  Y2M=YP1*YM1
  P(1)=-XM1*X2M/16.

```

```

P(2) = ((3.*XX+2.)*XX-9.)*XM1/16.
P(3) = -XX*XX/8.+1.125-P(2)
P(4) = XP1*XM/16.
Q(1) = -YM1*Y2M/16.
Q(2) = ((3.*YY+2.)*YY-9.)*YM1/16.
Q(3) = -YY*YY/8.+1.125-Q(2)
Q(4) = YP1*Y2M/16.
17 PX(4) = (0.375*XX-0.125)*XP1
   PX(1) = 0.5*XX-PX(4)
   PX(2) = (1.125*XX-1.375)*XP1
   PX(3) = -0.5*XX-PX(2)
   QY(4) = (0.375*YY-0.125)*YP1
   QY(1) = 0.5*YY-QY(4)
   QY(2) = (1.125*YY-1.375)*YP1
   QY(3) = -0.5*YY-QY(2)
18 DO 27 K=1,K4
   HP(K) = 0.0
   CP(K) = 0.0
   SP(K) = 0.0
19 HPX(K) = 0.0
   HPY(K) = 0.0
   CPX(K) = 0.0
   CPY(K) = 0.0
   SPX(K) = 0.0
   SPY(K) = 0.0
20 KK = ( (K+L)*KY+N ) *KX + M -KCON
   DO 23 J=1,4
   HD(J) = 0.0
   CD(J) = 0.0
   SD(J) = 0.0
21 HS(J) = 0.0
   CS(J) = 0.0
   SS(J) = 0.0
22 JJ = J*KX+KK
   DO 23 I=1,4
   II = I+JJ
   HT(I,J) = XHT(II)
   CT(I,J) = CSK(II)
23 ST(I,J) = SNK(II)
   DO 25 I=1,4
   DO 25 J=1,4
   HD(I) = Q(J)*HT(I,J)+HD(I)
   CD(I) = Q(J)*CT(I,J)+CD(I)
   SD(I) = Q(J)*ST(I,J)+SD(I)
   HS(I) = P(J)*HT(J,I)+HS(I)
   CS(I) = P(J)*CT(J,I)+CS(I)
   SS(I) = P(J)*ST(J,I)+SS(I)
25 CONTINUE
   DO 27 I=1,4
   HP(K) = HD(I)*P(I)+HP(K)
   CP(K) = CD(I)*P(I)+CP(K)
   SP(K) = SD(I)*P(I)+SP(K)
26 HPX(K) = HD(I)*PX(I)+HPX(K)
   CPX(K) = CD(I)*PX(I)+CPX(K)
   SPX(K) = SD(I)*PX(I)+SPX(K)
   HPY(K) = HS(I)*QY(I)+HPY(K)
   CPY(K) = CS(I)*QY(I)+CPY(K)
   SPY(K) = SS(I)*QY(I)+SPY(K)

```

```

27 CONTINUE
   H=0.0
   CK=0.0
   SK=0.0
28 HX=0.0
   HY=0.0
   CKX=0.0
   CKY=0.0
   SKX=0.0
   SKY=0.0
29 DO 31 K=1,K4
   H=C(K)*HP(K)+H
   CK=C(K)*CP(K)+CK
   SK=C(K)*SP(K)+SK
30 HX=C(K)*HPX(K)+HX
   HY=C(K)*HPY(K)+HY
   CKX=C(K)*CPX(K)+CKX
   CKY=C(K)*CPY(K)+CKY
   SKX=C(K)*SPX(K)+SKX
   SKY=C(K)*SPY(K)+SKY
31 CONTINUE
   RAD=SQRTF(CK*CK+SK*SK)
   CK=CK/RAD
   SK=SK/RAD
32 DKX=CK*SKX-SK*CKX
   DKY=CK*SKY-SK*CKY
33 RETURN
   END

```



```

SUBROUTINE SHIP
COMMON X,Y,A,B,CC,H,CK,SK,KX,KY,KT,FOX,FOY,KCON
COMMON/L2/HX,HY
COMMON/L4/AX,AY,BX,BY,CX,CY
SPEED PARAMETERS OF A AP3
FAC1=H+20.710888079
VH=FAC1**2.1496946258*EXP(-0.093470593044*H)*0.025341054963
FAC1=2.1496946258/FAC1
FAC2=FAC1-0.093470593044
DVH=VH*FAC2
FAC1=H+H.9454862274
VF=FAC1**0.34831219696*EXP(-0.025324954736*H)*7.9637908524
FAC1=0.34831219696/FAC1
FAC2=FAC1-0.025324954736
DVF=VF*FAC2
FAC1=H+37.097335211
H=FAC1**3.5824360581*EXP(-0.088029452490*H)*0.40848873291E-4
FAC1=3.5824360581/FAC1
FAC2=FAC1-0.088029452490
DR=B*FAC2
A=(VF+VH)*0.5
CC=A-VH
DA=(DVF+DVH)*0.5
DC=DA-DVH
AX=DA*HX
AY=DA*HY
BX=DR*HX
BY=DR*HY
CX=DC*HX
CY=DC*HY
RETURN
END

```

```

SUBROUTINE VDERIV
COMMON X,Y,A,B,CC,H,CK,SK,KX,KY,KT,FOX,FOY,KCON
COMMON/L3/DKX,DKY
COMMON/L4/AX,AY,BX,RY,CX,CY
COMMON/L5/CAPV,CAPVX,CAPVY,CAPVU,W
REAL MSF,MSFX,MSFY,NUM
DELX=X-FOX
DELY=Y-FOY
MSF=(DELX**2+DELY**2+973.75)/1043.638743
MSFX=DELX/521.8193715
MSFY=DELY/521.8193715
IF( LR .EQ. 0 )GO TO 1
COST=COSF(W)*CK + SINF(W)*SK
SINT=SINF(W)*CK - COSF(W)*SK
GO TO 2
1 COST=COSP*CK + SINP*SK
SINT=SINP*CK - COSP*SK
2 A2MC2=A*A-CC*CC
ROOT=SQRTF(B**2*(COST**2)+A2MC2*(SINT**2) )
NUM=R*A2MC2/A
DENOM=B*CC*COST/A+ROOT
V=NUM/DENOM
VGRID=V/8.5660416667
CAPV=VGRID*MSF
AMCX=A*AX-CC*CX
AMCY=A*AY-CC*CY
FX=(CC*RX+B*CX-R*CC*AX/A)/A
FY=(CC*RY+B*CY-R*CC*AY/A)/A
BPCMA2=R*B+CC*CC-A*A
TERMX=(COST*FX+B*CC*SINT*DKX/A+(AMCX*SINT*SINT+B*RX*COST*COST +
1 BPCMA2*SINT*COST*DKX)/ROOT)*(A*VGRID/(B*A2MC2) )
TERMY=(COST*FY+B*CC*SINT*DKY/A+(AMCY*SINT*SINT+B*RY*COST*COST
1 +BPCMA2*SINT*COST*DKY)/ROOT)*(A*VGRID/(B*A2MC2) )
VX=VGRID*(2.*AMCX/A2MC2 +BX/B-AX/A-TERMX)
VY=VGRID*(2.*AMCY/A2MC2+BY/B-AY/A-TERMY)
CAPVX=VX*MSF+VGRID*MSFX
CAPVY=VY*MSF+VGRID*MSFY
VU=VGRID*(B*CC*SINT/A+BPCMA2*SINT*COST/ROOT)/(B*CC*COST/A +ROOT)
CAPVU=MSF*VU
RETURN
END

```

```

SUBROUTINE ANGLE
COMMON X,Y,A,B,CC,H,CK,SK,KX,KY,KT,FOX,FOY,KCON
COMMON/L6/ XK,XLG,XLT
DELX=X-FOX
DELY=Y-FOY
COSXK=-DELX*CK-DELY*SK
SINXK=DELX*SK-DELY*CK
IF (COSXK) 2,1,2
1 XK=SIGNF(90.,SINXK)
GO TO 6
2 XK=ATANF(SINXK/COSXK)*57.29577951
IF (COSXK) 3,6,6
3 IF (SINXK) 5,4,4
4 XK=XK+180.
GO TO 6
5 XK=XK-180.
6 IF (XK) 7,8,8
7 XK=360.+XK
8 XT=DELX*.98480775-DELY*.17364818
YT=DELX*.17364818+DELY*.98480775
RAD=SQRTF(XT*XT + YT*YT)
IF (XT) 10,9,10
9 XLG=SIGNF(90.,YT)
GO TO 14
10 XLG=ATANF(YT/XT)*57.29577951
IF (XT) 11,14,14
11 IF (YT) 13,12,12
12 XLG=XLG + 180.
GO TO 14
13 XLG=XLG - 180.
14 XLT=-ATANF(RAD/31.205)*114.591559 + 90.
RETURN
END

```

IDENT DTGB  
 PROGRAM LENGTH  
 BLOCKS  
 PROGRAM\* LOCAL  
 ENTRY POINTS

000000 DTGB

EXTERNAL SYMBOLS

TRFDTG

```

      ENTRY DTGB
*      FORTRAN CALL TO THIS PROGRAM---CALL DTGB(A,B,C)
*      ---WHERE A = ADDRESS OF DTG WORD(LEFT JUSTIFIED)B1
*              B = ADDRESS TAU B2
*              C = ADDRESS NEW DTG WORD(LEFT JUSTIFIED)B3
*      RETURN WITH NEW DTG IN C
DTGB  BSSZ  1
      SX6  B3
      SA6  T
      SA1  R1
      LX1  48          RGT ADJUST DTG
      SA2  B2
      RJ   =XTRFDTG
      SA1  T
      SB3  X1
      LX6  12
      SA6  B3
      JP   DTGB
T      DATA 0
      END

```

UNUSED STORAGE

21 STATEMENTS

3 SYMBOLS

```

IDENT  FIELDSX
PROGRAM LENGTH
BLOCKS
PROGRAM*  LOCAL
ENTRY POINTS
000000 FIELDSX
EXTERNAL SYMBOLS
EXDEFIT  WAR64X  RDMST

ENTRY  FIELDSX

REVISED BY S.W. SEIFRIDGE JUN 1968 MELLONICS

SELECT FNWF FIELD FROM MASTER, UNPACK AND FLOAT DATA

FORTRAN CALL TO THIS PROGRAM--- CALL FIELDS(A,P,C,D)
---WHERE A = ADDRESS OF 3989 WORD BIN(CONTAINS FLTD DATA ON EXIT)
        B = ADDRESS OF DATE-TIME GROUP WORD
        C = ADDRESS OF PARAMETER SELECTION INDEX(0,1,2,3)
            WHERE 0 = CD,1 = CH,2 = U CURR,3 = V CURR
        D = ADDRESS OF READ-MASTER ERROR INDICATOR WORD
            (IF FIELD NOT FOUND OR PARITY ERROR (D)=1 ON RETURN)

FIELDSX  BSSZ  1
        SAVE FORTRAN CALL PARAMETERS
        SA2  B2
        SX6  B1
        RX7  X2
        SA6  PARAM
        SA7  TABLE+1
        SA3  B3
        RX6  X3
        SA6  COUNTER
        SX7  B4
        SA7  PARAM+1
        READ ONE FIELD FROM MASTER TAPE
        SA2  COUNTER
        SA1  NAMTAB+X2
        RX6  X1
        SA6  TABLE
        SB1  A6
        RJ  =XRDMST
        UNPACK FIELD
        SA0  B0
        SB1  PAF
        SA2  PARAM
        SB2  X2
        SB3  7601R
        PJ  =XWAB64X
        SHIFT DATA INTO LOWER 48 BITS
        SB1  20
        SB2  3988
        DTG
        STORE SELECTION INDEX
        NOT-FOUND INDICATOR
        BCD PARAMETER NAME OF FIELD
        LOC. UNPACKED FIELD

```



	SB3	B0	
	SB7	1	
	SA1	PARAM	
SHIFT	SA2	X1+B1	
	AX2	12	
	RX6	X2	
	SA6	X1+B3	
	SB1	B1+B7	
	SB3	B3+B7	
	GE	B2,B1,SHIFT	
*	CONVERT	FIELD TO FLOATING POINT	
	SB1	X1	
	SB2	X1	
	SA1	COUNTER	
	SA2	SCATAB+X1	SCALE FACTOR
	SB3	X2	
	SB4	7601R	
	RJ	=XFXDFLT	
	JP	FIELDSX	
* TARIF	DATA	0	
	DATA	0	
	VFD	30/PAF,30/ERR	
	DATA	5LTAPE4	
	DATA	3LEEE	
NAMTAR	DATA	10HCD	
	DATA	10HCH	
	DATA	10HU CURR	
	DATA	10HV CURR	
SCATAB	DATA	41	
	DATA	41	
	DATA	36	
	DATA	36	
COUNTER	DATA	0	
PARAM	RSSZ	2	
ERR	SA1	PARAM+1	
	SB1	1	
	SX6	B1	
	SA6	X1	
	JP	FIELDSX	
PAF	RSSZ	1100	
	END		

UNUSED STORAGE

85 STATEMENTS

13 SYMBOLS

IDENT SEALAND  
PROGRAM LENGTH

BLOCKS

PROGRAM\* LOCAL

ENTRY POINTS

000000 SEALAND

EXTERNAL SYMBOLS

LANDSEA

ENTRY SEALAND

FORTTRAN CALL TO THIS PROGRAM--- CALL SEALAND(I,J,LAND,KER)

-WHERE I = ADDRESS FO REAL I INDEX ON FNWC 63X63 GRID

J = ADDRESS OF REAL J INDEX

LAND = ADDRESS OF LAND/SEA INDICATOR (0 FOR SEA, 1 FOR LAND)

KER = ADDRESS OF ERROR INDICATOR (OUT OF BOUNDS)

(0 IF ERROR ABSENT, 1 IF ERROR IS PRESENT)

```
SEALAND BSSZ 1
SA1 R1 I
SA2 R2 J
UX1 X1,R1
UX2 X2,R2
SH1 R1+39
SH2 R2+15
LX1 X1,R1 I S39
LX2 X2,R2 J S15
MX0 36
RX1 X0*X1
RX2 -X0*X2
RX1 X1+X2 I BITS 24-47, J BITS 0-23,
SX6 R3 EACH SCALED S15
SX7 R4
SA6 LSIN SAVE L/S IND. ADDRESS
SA7 ERRIN SAVE ERROR IND. ADDRESS
SH1 SEA
SH2 LAND
SH3 ERROR
JP =XLANDSEA
SEA MX6 0
MX7 0
JP OUT
LAND SX6 1
MX7 0
JP OUT
ERROR SX6 1
SX7 1
OUT SA1 LSIN
SA2 ERRIN
SA6 X1
SA7 X2
JP SEALAND

LSIN BSSZ 1
ERRIN BSSZ 1
FN()
```

DIMENSION OF WAVE FIELD ARRAYS=( 22, 32, 12)      I MIN= 10      J MIN= 12  
 ROUTE OF SHIP BEGINS    0 HOURS AFTER TIME SERIES ORIGIN  
 FROM LONGITUDE = 154.0 AND LATITUDE = 41.0  
 TO LONGITUDE = -123.0 AND LATITUDE = 38.0

DAYS OF TRAVEL	GEODESIC ROUTE			
	LONGI- TITUDE	LATI- TITUDE	WAVE HEIGHT	WAVE DIRECTION FROM NORTH
0.00	154.0	41.0	21.21	319
.50	156.6	42.0	19.31	304
1.00	159.9	43.0	16.76	231
1.50	163.8	44.2	15.63	252
2.00	167.5	45.1	20.00	270
2.50	171.0	45.8	19.54	199
3.00	175.3	46.5	16.96	266
3.50	179.6	47.1	11.64	219
4.00	-175.4	47.5	9.71	109
4.50	-170.3	47.7	10.36	208
5.00	-165.2	47.7	4.09	121
5.50	-160.0	47.4	6.55	232
6.00	-155.0	47.0	9.48	150
6.50	-150.1	46.3	11.78	175
7.00	-145.5	45.4	17.28	176
7.50	-141.2	44.4	13.41	162
8.00	-136.8	43.2	7.39	153
8.50	-132.6	41.8	4.86	162
9.00	-128.5	40.2	1.91	29
9.50	-124.6	38.5	6.44	346
9.70	-123.1	37.8	13.70	314

# MINIMUM TIME ROUTE

DET R = .00				
DAYS OF TRAVEL	LONGI- TUDE	LATI- TUDE	WAVE HEIGHT	WAVE DIRECTION FROM NORTH
0.00	154.0	41.0	21.21	319
.48	157.7	40.4	14.80	1
.94	161.8	40.3	6.92	171
1.44	166.2	40.5	10.10	291
1.92	170.2	40.9	14.70	159
2.40	174.3	41.6	15.68	281
2.88	178.1	42.7	17.50	280
3.36	-178.0	44.0	13.88	275
3.84	-173.8	45.1	12.69	282
4.32	-169.3	45.9	12.48	275
4.80	-164.7	46.0	7.06	142
5.28	-160.0	45.6	6.89	334
5.76	-155.6	44.9	9.35	147
6.24	-151.3	44.3	10.78	157
6.72	-147.3	43.7	13.81	164
7.20	-143.4	43.1	12.97	171
7.68	-139.3	42.5	9.42	166
8.16	-135.1	41.7	5.62	296
8.64	-130.8	40.7	3.96	115
9.12	-126.8	39.4	1.29	1
9.60	-123.0	38.1	15.20	8

# WEIGHTED TIME ROUTE

DET R = .92				
DAYS OF TRAVEL	LONGI- TUDE	LATI- TUDE	WAVE HEIGHT	WAVE DIRECTION FROM NORTH
0.00	154.0	41.0	21.21	319
.48	157.6	39.9	13.39	30
.97	161.6	39.6	5.18	145
1.45	166.0	39.8	9.10	275
1.94	170.2	40.4	13.90	271
2.42	174.2	41.5	15.39	281
2.90	177.9	42.9	17.61	281
3.39	-178.4	44.4	13.48	270
3.87	-174.3	45.9	11.71	275
4.36	-169.7	46.8	11.14	264
4.84	-164.9	47.0	5.44	136
5.32	-160.1	46.4	6.41	309
5.81	-155.6	45.5	9.76	148
6.29	-151.3	44.7	10.94	158
6.78	-147.4	43.8	14.09	165
7.26	-143.5	43.1	12.77	171
7.74	-139.4	42.4	9.14	167
8.23	-135.1	41.7	5.74	187
8.71	-130.9	40.7	3.73	109
9.20	-126.8	39.5	1.17	359
9.68	-122.9	38.2	15.90	183

## GLOSSARY

### TEST

IMIN JMIN	= Minimum indices of wave height data subfield
KX, KY, KT	= Dimensions of wave data subfield
DTG(KT)	= Array of octal-date time groups
U(200), DELU(200)	= Control arrays
GRADX, GRADY, GRADZ	= Gradient arrays
AZ, AL, HT, WA	= Arrays for track position tabulation
WE(3989), WD(63,63)	= Unpacking area of core
XLAM, XMU, XSIGMA	= Components of adjoint vectors $\Lambda^1, \Lambda^2, \Lambda^3$
X,Y	= Rectangular coordinates of ship's position relative to subfield origin
A,B,CC	= Parameters for elliptical polar velocity diagram Fig. 1
H, CK, SK	= Wave height, cosine and sine of wave direction relative to OXY axes
FOX, FOY	= Floating point coordinates of North Pole relative to subfield origin
XHT, CSK, SNK	= Wave field arrays
T	= Time in days from first member of time series
TFS	= Time in days from beginning of track
XMEAN, YMEAN	= Coordinates of midpoint of subfield
ROX, ROY	= Semi-dimensions of a rectangle in the subfield within which track must lie



LR	= Index to denote a geodesic track (LR=0) or an optimum track (LR=1)
COSLG1, SINLG1	= Cosine and sine of longitude of initial point
COSLT1, SINLT1	= Cosine and sine of latitude of initial point
COSLG2, SINLG2	= Cosine and sine of longitude of terminal point
COSLT2, SINLT2	= Cosine and sine of latitude of terminal point
X1, Y1, X2, Y2	= Coordinates of initial and terminal points of track relative to North Pole in grid units
S12	= Straight line distance from X1, Y1, to X2, Y2
ARC	= Great circle distance from X1, Y1 to X2, Y2
XSTART, YSTART	= Coordinates of initial point in grid units relative to subfield origin
XEND, YEND	= Coordinates of terminal point in grid units relative to subfield origin
COSP, SINP	= Cosine and sine of ship's heading at any point along great circle track
V	= Control program to define initial control $u(t)$
GAMA	= Weighting factor used in calculating optimum track, GAMA=0 minimum time, GAMA > 0 weighted time track
DELUMX	= Maximum allowable change in the control $u(t)$ at any instant of time
N, L	= Counters to stop computation if convergence is questionable

STEP	= Variable time step of integration
XDOT, YDOT, ZDOT	= Time derivative of state variables
MSF	= Map scale factor for stereographic projection
A11, A12, A22	= Elements of A matrix of Section III
Z11, Z12, Z13, Z22, Z23, Z33	= Elements of Z matrix of Section III
DET	= Determinant of A matrix of Section III
C1, C2	= Elements of C matrix of Section III
E1, E2, DELTAU	= Elements of E matrix of Section III
NN	= Index to denote a minimum time track (NN=0) or a weighted time track (NN=1)
B11, B12, B13, B22, B23, B33	= Elements of B matrix of Section III
D	= Determinant of B matrix
D1, D2, D3	= Elements of D matrix of Section III
B1, B2, B3	= Elements of $\alpha$ matrix of Section III
RR	= Coefficient of DELU(T) to limit maximum change in control at any time to DELUMX
TERP	
C(4)	= Weighting factors for interpolation in the time demension between K4 ordinates
L	= Index to determine member of field time series
TT	= T of Eq. (20) of reference [5]

M, N	= Indices to pick out x, y grid point data
XX, YY	= XY of equation (18) of reference [4]
P(1) to P(4)	= P1 to P4 of equation (19) of reference [4]
Q(1) to Q(4)	= Elements of $P^T(y)$ matrix of equation (18) of reference [4]
PX, QY	= Partial derivatives of P(1) to P(4) and Q(1) to Q(4)
HX, HY, CKX, CKY, SKX, SKY	= Partial derivative of H, cos K, sin K
DKX, DKY	= Partial derivatives of wave direction angle K

#### SHIP

VH, VF	= Ship's speed in knots in head waves and following waves
AX, BX, CX, AY, BY, CY	= Partial derivatives of parameters of elliptical polar velocity diagram

#### VDERIV

MSFX, MSFY	= Partial derivatives of map scale factor
COST, SINT	= Cosine and sine of the angle $\theta$ of Figure 1
VGRID	= Ship's speed in knots relative to the stereographic grid
CAPV	= Ship's speed in knots relative to the earth's surface
CAPVX, CAPVY, CAPVU	= Partial derivatives of CAPV

# INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2. Library Naval Postgraduate School Monterey, California 93940	2
3. Prof. William E. Bleick Code 53BL Naval Postgraduate School Monterey, California 93940	10
4. Prof. Frank D. Faulkner Code 53KD Naval Postgraduate School Monterey, California 93940	10
5. Lt. John D. Carpenter, USN Country Club Drive Carmel Valley, California 93924	1
6. Capt. George D. Gregor, USN (Ret.) 74 Woodman's Trail Wakefield, Rhode Island 20475	1
7. Lt. Richard A. Gregor, USN 74 Woodman's Trail Wakefield, Rhode Island 20475	10





UNCLASSIFIED

Security Classification

## DOCUMENT CONTROL DATA - R &amp; D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940		2a. REPORT SECURITY CLASSIFICATION Unclassified	
		2b. GROUP	
3. REPORT TITLE OPTIMUM SHIP ROUTING BY THE METHOD OF STEEPEST ASCENT			
4. DESCRIPTIVE NOTES (Type of report and, inclusive dates) Thesis			
5. AUTHOR(S) (First name, middle initial, last name) GREGOR, RICHARD ALLEN, Lieutenant, USN			
6. REPORT DATE September 1968		7a. TOTAL NO. OF PAGES 53	7b. NO. OF REFS 6
8a. CONTRACT OR GRANT NO.		9a. ORIGINATOR'S REPORT NUMBER(S)	
b. PROJECT NO.			
c.		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.			
10. DISTRIBUTION STATEMENT <del>This document is subject to special export controls and each transmission to foreign government or foreign nationals may be made only with prior approval of the Superintendent, Naval Postgraduate School, Monterey, California 93940.</del>			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940	

## 13. ABSTRACT

With the advent of the high speed digital computer, many problems heretofore considered unsolvable for all practical purposes are now well within the reach of the applied mathematician. One such problem is the routing of a ship through a time dependent ocean wave field, from one point on the earth's surface to another, so as to minimize a cost function of the form  $g(x,y,t,u)$ .

This paper considers a numerical solution to the above problem. The technique to be employed is known as the method of steepest ascent and is attributed to Arthur E. Bryson and Walter F. Denham [1]. Although the computer program as given in the Appendix is written specifically for a VC2AP3 class vessel operating in a described area of the North Pacific Ocean, it can be readily modified to accommodate any type vessel operating in the Northern Hemisphere.

14

KEY WORDS

LINK A

LINK B

LINK C

ROLE

WT

ROLE

WT

ROLE

WT

OPTIMUM SHIP ROUTING

CALCULUS OF VARIATIONS

STEEPEST ASCENT

NO FORN

NO FORN











thesG758

DUDLEY KNOX LIBRARY



3 2768 00414788 4

DUDLEY KNOX LIBRARY